

Milestone 5 Report

一、引言

1.1 编写目的

本文档的编写旨在总结此次pytorch项目分析平台的开发工作，对整个项目的开发过程、目的、产品、意义进行评价，对人员的分配、工作情况进行分析总结，以提升团队的项目开发能力，促使团队成长。

1.2 项目名称

pytorch项目分析平台

1.3 相关人员

- 任务提出者：浙江大学软件需求工程任课老师：
- 开发者：浙江大学软件需求工程课程xyyz小组: 陈奕宇、黄亦霄、江小小
- 用户：开发者、代码分析者

二、项目概述

2.1 项目背景

本项目是浙江大学 2022-2023 学年秋冬学期的《软件需求工程》课程项目，目标为优化一个开源项目分析平台。在其项目信息检索模块，更新频率折线图模块，贡献比例饼状图模块，issues 柱状图模块这四个已经实现的模块下进行一系列优化处理。

2.2 项目介绍

本项目专注于pytorch项目的分析，以对pytorch项目的分析为核心构建分析平台。

- 平台：界面简洁大方，有各种图表来清晰展示数据。并且提供用户指南，来帮助用户更好的使用我们的平台来分析。
- 分析者：可以从多个维度了解pytorch项目，例如社区发展速度、贡献者活跃情况、核心贡献者贡献情况、issue分布情况等等。并且可以将pytorch和pandas项目进行对比。

2.3 产品成果

2.3.1 产品

最终成果是一个pytorch项目分析平台，包括

- 前后端代码
- 数据库建立代码

- 系统开发文档

2.3.2 手册

- 配置安装手册
 - 提供给感兴趣的开发者、系统管理员和维护人员进行阅读，对于网站的部署方法进行了详细的说明。
- 开发手册
 - 提供给开发者、维护人员进行阅读，详细描述了后端api的使用方法，使他们可以更加方便的阅读源码，进行继续开发或是修复已有的问题

2.3.3 服务

提供使用教程及咨询服务，以解决用户在使用网站中的困难。提供后续一定年限的维护服务，当软件出现问题时，提供技术支持。

三、实际开发结果

3.1 主要功能

3.1.1 初始化模块

各类用户都可以通过访问特定的url来初始化原本为空的本地数据库。该模块自动将开发者获取到并缓存的数据输入数据库，无需用户手动导入数据。此后使用Pytorch分析平台便不再需要这一初始化模块，可直接进行使用。

3.1.2 分析信息显示模块

各类用户可以查看Pytorch项目贡献者的活跃情况及社区的发展速度；统计一段时间内来自不同贡献者的代码提交数，从绘制的词云图中得知Pytorch项目的核心贡献者（以commit数量衡量）；统计设计相关的讨论，即一些关于代码、可维护性、鲁棒性的Pull Request，并分析可视化设计讨论的数量是如何随时间变化的，以及其中核心贡献者的贡献率；对 PyTorch 项目的 stargazer, committer, issue 人数的 company 信息分别进行数据可视化分析，通过气泡图形式实现，直观体现其占比大小等不同。以上信息均可以pdf形式导出。

3.1.3 项目对比模块

各类用户可以横向对比均为热门开源项目的Pytorch项目和Pandas项目的上述各种分析信息。

3.1.4 数据更新模块

由于本项目缓存的数据以管理员更新并写入csv文件为主，各类用户获取到的项目文件不一定缓存的是最新的数据，因此若各类用户希望看到关于Pytorch项目和Pandas项目的最新数据，可以通过点击刷新按钮动态获取并显示最新的上述各种分析信息。

3.2 开发流程

3.2.1 项目组织结构

项目总负责人——陈奕宇 统筹整个项目工程的实现，领导项目小组完成全部项目工作内容并进行后端开发以及数据库设计。

前端开发 ——江小小 建立系统框架；概要设计；前后端分离；前端界面搭建；参加技术评审。

后端开发 ——黄亦霄 负责进行编码工作与单元测试，进行系统集成，及时解决测试时出现的问题。

测试工作 ——陈思全 编写测试方案。

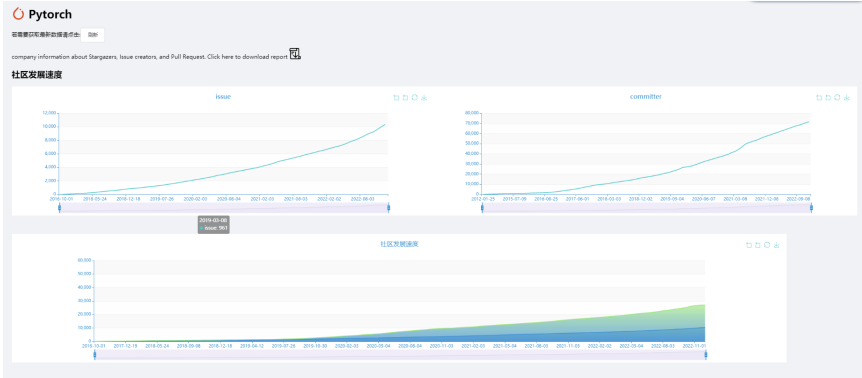
文档工作 ——姚逸飞 记录项目流程。

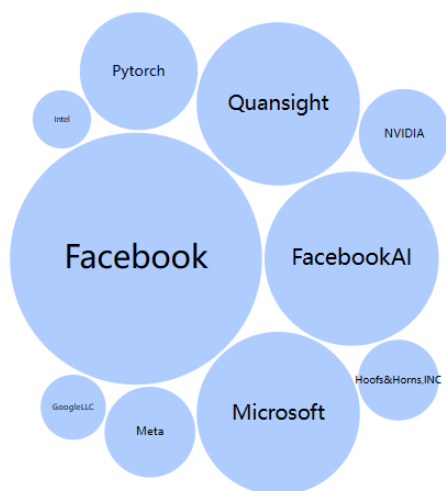
3.2.2 项目阶段划分及关键任务

项目阶段	持续时间	主要工作	输出内容
Milestone1	2022.09.12–2022.10.07	选择项目，进行项目可行性分析	《Milestone 1 Report》
Milestone2	2022.10.08–2022.10.21	进行愿景、范围分析，绘制项目路线图	《Milestone 2 Report》
Milestone3	2022.10.22–2022.11.11	进行需求分析，确定项目功能，制定项目计划	《Software Requirements Specification》、Mid-term Presentation
Milestone4	2022.11.12–2022.12.16	进行系统设计和编码	描述过程的视频
Milestone5	2022.12.17–2022.12.23	进行系统测试，项目总结	《Milestone 5 Report》、Final Presentation

3.3 项目展示截图

3.3.1 Pytorch分析界面





3.3.2 项目对比界面



四、开发工作评价

4.1 生产效率

- 项目启动于2022年9月，结束于2022年12月，项目编码开发阶段启动于2022年10月，
- 整个项目及编码开发阶段均经历了三个月。
- 程序的平均生产效率：1200行/(月·人)
- 文档的平均生产效率：1千字/(月·人)
- 评价：优秀。项目严格按照《项目计划》中的时间线一步步推进，及时甚至提早达到里程碑要求，代码编写规范、功能开发齐全；文档编写完备，有规范可循。

4.2 产品质量

- 设计：系统遵守MVC的设计价值观，提供了良好的用户体验，表现在界面简洁美观，易用性高，提示性强等方面。
- 功能：系统实现了用户代表所提出的基本需求，并在需求变更的情况下对需求与用例进行了适当的增删使之更好地符合用户的需求。通过测试可知，产品在功能上没有较大偏差，且通过了安全性测试、压力测试等非功能性需求。
- 性能：前后端支持多并发，查询刷新时延的典型值在100ms以内，动态图表交互没有可感延迟。后端系统部署时间在20min内（未上线的新版本后端初始化器可压缩至1min）。
- 文档：文档按照项目章程中的计划完成，文档齐全、规范。图表完整清晰，较好地反映了整个项目的设计思路、架构，且明白易懂，较好地辅助共同交流。
- 评价：优秀。产品在功能性、非功能性需求上均取得了良好的表现。

4.3 技术选型

- 开发：采用前后端分离的架构，前端使用Vue.js，用组件化的解决方案提升了开发效率。后端使用Django+MySQL，开发社区健全，易于查找解决方案。使用git/GitHub版本控制工具进行前后端分离开发与进度管理。
- 管理：分布式管理，每名成员负责一个如“前端”，“后端”，“文档”等模块的管理工作，各自对项目负责。在涉及模块对接时由对应模块的负责同学进行。
- 评价：优秀。使用了经过时间检验的框架与经久不衰的管理技术。

五、经验与教训

- 开发阶段需要个人探索和统筹规划，二者不可偏废。本组开发过程中后端模块出现过三个独立大版本A/B/C和一个小版本d，最终结合实际功能完成程度选型ABd，不仅造成了一定的浪费，还加重了后端代码的臃肿。
- 工程管理能力还有进步空间。本组后端模块在首轮访问GitHub API时取得的数据难以使用，但由于过程极为耗时，只能放弃该模块换用其他技术路线。今后开始较为耗时的工作前需要交叉检查配置。
- 前后端对接时应保证对接双方设备配置均达标。本组前后端对接时，后端开发同学起初无法运行前端服务器，占用了大量对接时间排除故障，影响合作效率。今后技术选型应保证所有开发环境运行正常之后再开始。